Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Lab 2: Digital Modulation & Transmission

1. Overview

In this lab, we will employ fully digital communication system, exploring common forms on digital carrier modulation, ideal receiver structure, bit-error performance and understand the impact of noise and interference on the performance of digital communication system. In the later part, we all also explore carrier synchronisation and symbol synchronisation (timing recovery) in an example scenario to explain the need and importance of these blocks in a modern digital system [and in turn, how SDR could play a role in such systems]. Theoretical backgrounds, if any, will be covered in the appendix

1.1. Outcomes

Upon successful completion of this laboratory, the student will:

- Identify and characterise common forms of digital carrier modulation using time-domain and frequency-domain observations
- Evaluate the bit error performance of a digital communication system and compare to theoretical values derived from the model
- Understand the influence of noise and interference on the performance of a digital communication system.
- Understand the importance of carrier and symbol synchronisation (timing recovery) in a digital communication system.

1.2. Background

Appendix A provides a summary of mathematical models for amplitude-shift keying (ASK), phase-shift keying (PSK) ad frequency-shift keying (FSK) digital modulation schemes.

2. Digital carrier modulated waveforms

First, we will use your individual Pluto SDR devices to observe digital carrier modulated communication signals. For the purpose of this lab, we have a transmitter setup using a Pluto SDR that broadcast radio-frequency signals in the laboratory at a frequency in the 865 MHz frequency band.

The broadcast consist of multiple communication signals, each having a distinct carrier frequency and frequency band (i.e., frequency division multiplexing is used by the single transmitting device). The carriers are separated by approximately 30 kHz. The communication signals are binary ASK, binary FSK and binary PSK.

In the first step, you will build a simple receiver with a spectrum analyser and a time-scope to observe the communication signals in both time and frequency domain.

Question 2.1: Take a screen capture of the Spectrum Analyzer window showing the broadcast signal that is observed

Question 2.2: Identify each of the digital communication signals within your received signal. From the screen captures of Spectrum Analyzer and Time Scope, can you characterise each signal? Also, from the time/frequency plots, can you determine the bit rate for each signal and indicate how you determined them. Compare, and contrast the salient features seen in the plots.

Question 2.3: One of communication signals in your received wide-band signal use binary Amplitude Shift Keying (ASK), which is essentially a binarised amplitude modulated (DSBAM) waveform. Using your non-coherent DSBAM receiver from Lab 1, recover the actual baseband message. Add a screen capture of the Simulink model and a Time-Scope window showing the message waveform in your submission.

NOTE: You will have to adapt the configurable parameters of receiver, decimation factor and filter block according to the spectrum you observe.

---

3. Bit Error Rate Simulation

NOTE: This section does not require you to use the PLUTO device or the transmitter in the lab, hence, come back to this section at the end of part 4.

In this section, we will simulate the bit error performance of a binary phase-shift keying communication system and compare its results with the theoretical prediction for the probability of bit error in a BPSK system.

Recall that for equiprobable symbols, the probability of symbol (or equivalently, the probability of bit error in case of binary modulation) is give by

$$P_{e,BPSK} = Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$$

where E_b is the average energy per bit and $Q(\cdot)$ is the Q-function. We make the assumption that the received waveform is corrupted by an additive white Gaussian noise (AWGN) process with zero mean and power spectral density of $N_o/2$, assuming an optimal receiver [We do not cover this in theory in 4C21, but you should have covered this in 3C5 Telecommunications module].

The script provided with this lab-sheet (`plot_Ph_vs_EbNo_BPSK.m`) calculates and plots the bit-error probability of BPSK as a function of E_b/N_o.

To simulate the BPSK communication, create a Simulink model as shown in the Fig 1. The blocks used are:

- The **Bernoulli Binary Generator** block generates random binary numbers using a Bernoulli distribution. This block acts as our message source in the simulation. Set the Sample Time to 0.001, which gives a bit rate of 1000 bits per second. Set the Samples per frame to 1000, which specifies that the model will operate using vectors (or frames) containing 1000 bits [1].
-
- The **BPSK Modulator Baseband** block applies BPSK modulation to the message bits, producing the baseband representation of the communication signal.
-
- The **Ideal Rectangular Pulse Filter** translates the baseband symbols into rectangular pulse shapes. Set the `Pulse length` to 8, which will produce a rectangular pulse 8 samples long for each BPSK symbol. Therefore, the block increases the signal sampling rate by a factor of 8. Ensure the `Input processing` is configured for `Columns as frames` and `Rate options` is configured for `Enforce single-rate processing`.

[1] To learn more about how Simulink uses frames, see the documentation at https://www.mathworks.com/help/dsp/ug/sample-and-frame-based-concepts.html and https://www.mathworks.com/help/dsp/ug/inspect-sample-and-frame-rates-in-simulink.html

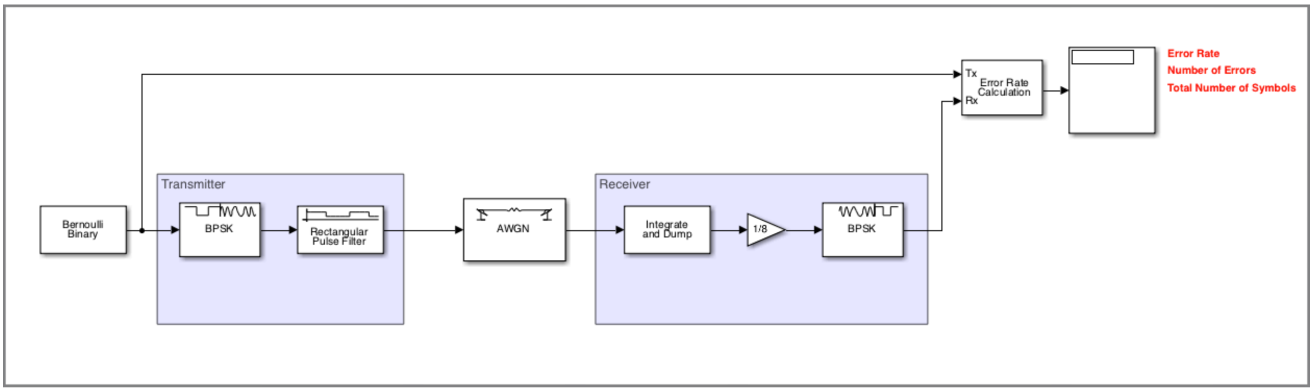

Fig 1: Simulink model to simulate BPSK communication

- The **AWGN Channel** block adds white Gaussian noise to the input signal, thus modelling the effects of a noisy communication channel. Set the `Symbol period` to 1/1000. Set the `Eb/No` parameter to 15.

- The **Integrate and Dump** block implements the ideal correlation receiver, or equivalently the ideal matched filter, for rectangular pulse shapes. Set the `Integration period` to 8 samples, which configures the block to integrate each received pulse (8 samples long) and produce an output value corresponding to each pulse.

- The **Gain** block, set to 1/8 normalises the output value to ±1 (without noise).

- The **BPSK Demodulator Baseband** block demodulates the input using BPSK. For each input value, the block decides which BPSK was sent and outputs the corresponding binary number.

- The Error Rate Calculation block compares the message data at the input to the transmitter to the received message data at the output of the receiver. Set `Output data` to Port and attach to a **Display** block. This configuration will display the error rate, the number of errors, and the total of symbols that were compared.

Experiment with the model by adding **Time Scope**, **Spectrum Analyzer**, **Eye Diagram**, and **Constellation Diagram** blocks. Be sure you understand how the model operates before proceeding.

Hint: You may find it helpful to change the Time Scope to a "stem" plot by selecting `View —> Style` and changing the `Plot type` setting.

Question 3.1: With a Time Scope block added, use it to display the output of the Ideal Rectangular Pulse Filter block. Can you determine the average signal power?
Question 3.2: Using a Time Scope block, observe the input to the AWGN Channel block, the output of the AWGN Channel block, and the output of the Gain block. In the AWGN Channel block, set the Input signal power to the value you calculated in the previous problem and set E_b/N_o to 12 dB. Set the remaining parameters based on your understanding of the simulation. Run the simulation, and take a screen capture of the Time Scope block. Explain and interpret the plots.
Question 3.3: Add Eye Diagram and Constellation Diagram blocks to the signal at the input of the BPSK Demodulator Baseband block. Set the AWGN Channel block for E_b/N_o of 12 dB. Run the simulation, and take screen captures of the eye diagram and signal constellation. Repeat for E_b/N_o of 6dB. Comment on the observations and how the plots change as a function of E_b/N_o.

Question 3.4: You will now use the model to simulate the system performance over a range of E_b/N_o values.

- Run the simulation for E_b/N_o values of 0dB to 12dB in 4dB steps.
- Set the simulation time to ensure the results are meaningful. This is especially important when very few bit errors are expected.
- Observe and record the error rate for each run.
- Plot bit error rate vs E_b/N_o and compare to the theoretical prediction. What are your thoughts on the results?

4. Carrier Synchronisation and Symbol Timing Recovery in BPSK

The simulation in Section 3 uses a baseband representation of a BPSK communication system. That is, the process of modulating the message information onto a high frequency sinusoidal carrier, as is typically required in wireless communication, was omitted from our simulation. As we discussed in the lecture, the frequency up-conversion process (translating the baseband message to a high-frequency carrier using a mixer) done in the transmitter can be exactly undone by the frequency downconversion process (translating the message information imparted on the high-frequency carrier down to baseband using the same mixer) in the receiver. Therefore, our simulations are valid for wireless systems under the assumption that the frequency translation processes are ideal.

For coherent communication systems, the key challenge is to ensure synchronisation between the transmitter and receiver. In this part of the laboratory you will observe and investigate coherent digital modulation techniques using real-world wireless signals.

In particular, you will observe:

- Carrier Synchronisation: the process of ensuring that the receive carrier is synchronised, in both frequency and phase, to the transmit carrier.
- Symbol Timing Recovery: the process of ensuring that the receiver precisely knows the symbol timing so that it may determine when to sample the recovered message for making symbol decisions.

Both of these processes are essential for coherent communication. Symbol timing recovery, but not carrier synchronization, is required in non-coherent communication systems.

Since we are using SDRs, these processes are implemented through adaptive algorithms in software. Traditional radios use hardware circuitry to achieve the functionality. Our focus will be on observing and understanding (and seeing!) the importance of these processes within a coherent digital communication system, rather than the algorithms or circuits through which they are implemented. If interested in learning about these algorithms, you are encouraged to check detailed materials in the references of the course textbook.

Download the BPSK Receiver Simulink model from blackboard. The model is shown in Figure 2. The blocks in the model have been pre-configured for this laboratory exercise.

IMPORTANT: For the BPSK receiver to work properly, the carrier frequencies of the transmitter and receiver must be relatively close to each other (ideally, no more than a few hundred Hz to perhaps 1kHz apart). You may have to compensate the center frequency of the PLUTO-SDR Receiver block in the model for accurate results. For our experiment, this can be accomplished either the Frequency Correction (ppm) setting, while with a calibration process, you can de-tune the center frequency according based on the observations during the calibration run (usually involving a known stable frequency source like a Pilot symbol in Digital Television broadcast [in the US]) . Doing so provides a coarse frequency correction. A fine frequency correction will be

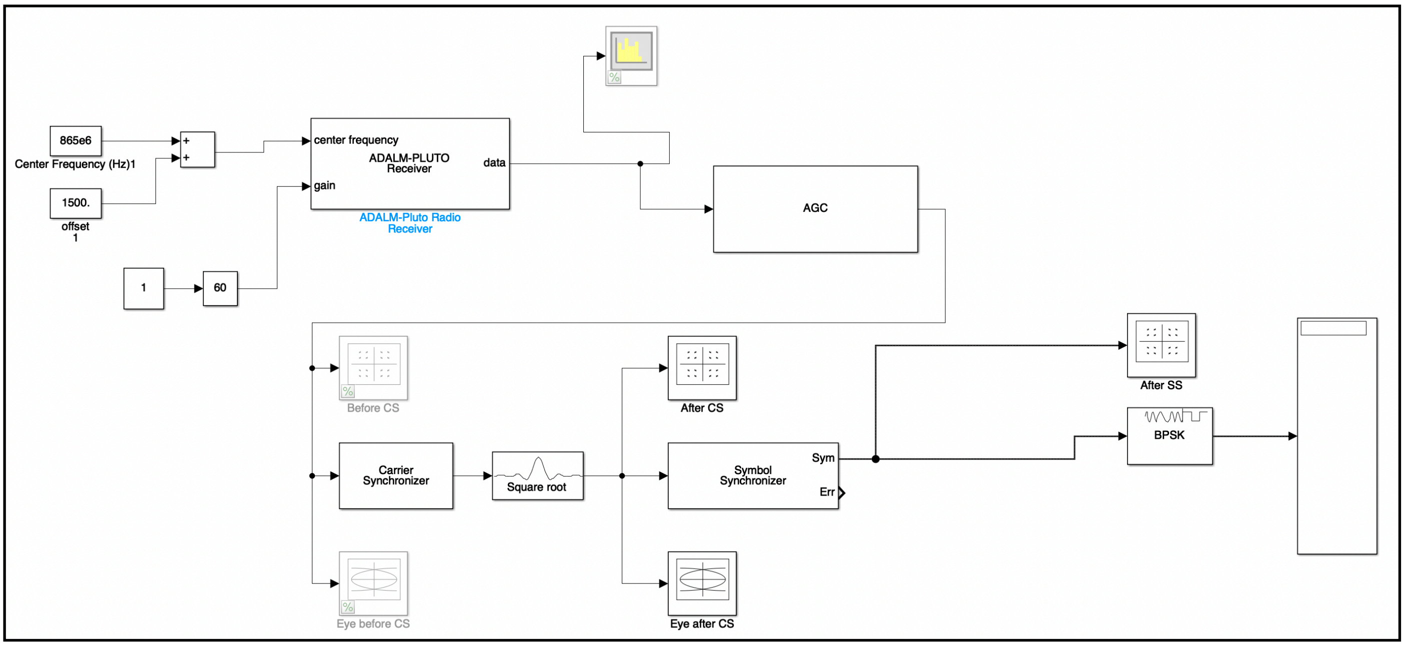

Fig 2. Simulink model for a BPSK receiver using PLUTO

applied through an adaptive algorithm in the Simulink model. Outside of this, do not modify any other (advanced) configuration parameter of the PLUTO-SDR Receiver block.

It is given that:

- The BPSK symbols are being transmitted at 20000 bits per second.

- The receiver model performs coarse frequency correction using the Frequency Correction (ppm) setting of the PLUTO-SDR Receiver block.
    - Sampling rate: 300e3
    - Output data type: single
    - Samples per frame: 55200

- The receive gain is controlled via a **Slider Gain** block, making it easy to adjust the receive gain while the model is running.

- The **AGC** block implements an automatic gain control system. This block adaptively adjusts its gain to give a constant signal level at its output. Such a block is common in systems in which the receiver input signal level changes due to variation in the communication channel.

- The **Carrier Synchronizer** block adjusts for carrier frequency and phase offsets in the incoming signal. This block will compensate for offsets that remain after coarse calibration of the carrier frequencies. It is be configured to observe the data with 15 samples per symbol, with damping factor 0.707 and loop bandwidth of 0.01. Modulation type is BPSK with phase left to auto.

- The **Raised Cosine Receive Filter** block filters the input signal using a raised cosine pulse shape filter. The block is configured to match the pulse shape used in your instructor's transmitter, and therefore mimics a matched filter. The block performs a decimation of 3.

- The **Symbol Synchronizer** block adjusts for symbol clock drift in the incoming signal, ensuring that incoming pulses are sampled (for purposes of deciding which symbol was sent) at the "opening" of the eye. The block is configured to output exactly one sample per symbol interval.

The block hence also performs decimation by 5.

- The **BPSK Demodulator Baseband** block demodulates its input signal and outputs the recovered bit sequence, which can be viewed on the Display block.

Question 4.1: Observe the Eye Diagram plots at the input and output of the Carrier Synchronizer block. Using a screenshot of the plots, explain any observations and/or interpretations.

Question 4.2: Observe the Constellation Diagram plots at the input and output of the Carrier Synchronizer block. Using a screenshot of the plots, explain any observations and/or interpretations.

Question 4.3: Observe the Constellation Diagram plots at the input and output of the Symbol Synchronizer block. Using a screenshot of the plots, explain any observations and/or interpretations.

Question 4.4: Observe the effects of reduced signal-to-noise ratio (SNR) at the input to the receiver by

- decreasing the Gain of the PLUTO-SDR Receiver
- placing an object near/over the receive antenna

Examine the effects of reduced SNR on Constellation Diagram and Eye Diagram plots. Take screen capture of your observations and provide any inferences you may have.

Submission

Submit the following for your observation as a short report via blackboard:

- Screenshot/captures and notes from all sections
- Screen capture of the signals, constellations accordingly
- Observations from simulating the digital communication system (section 3)
- Thoughts on questions 2.1 through 4.4

Appendix A: Binary Digital Carrier Modulation

1. Amplitude-shift keying (ASK), also known as on-off keying (OOK), can be represented as

$$s(t) = Am(t)\cos(2\pi f_c t) \tag{1}$$

where $m(t)$ is a baseband signal that takes on values of either 0 or +1 during each bit interval. The complex envelope is

$$g(t) = Am(t) \tag{2}$$

2. Binary phase-shift keying (BPSK) can be represented as

$$s(t) = \begin{cases} A\cos(2\pi f_c t), & \text{for t in the time interval when a 0 is sent} \\ A\cos(2\pi f_c t + \pi), & \text{for t in the time interval when a 1 sent} \end{cases} \tag{3}$$

$$= Am(t)\cos(2\pi f_c t) \tag{4}$$

where $m(t)$ is a baseband signal that takes on values ± 1 during each bit interval. The complex envelope is

$$g(t) = Am(t) \tag{5}$$

3. Frequency-shift keying (FSK) can be represented as

$$s(t) = \begin{cases} A\cos(2\pi f_{c_1} t), & \text{for t in the time interval when a 0 is sent} \\ A\cos(2\pi f_{c_2} t), & \text{for t in the time interval when a 1 sent} \end{cases} \tag{6}$$

$$= Am(t)\cos(2\pi f_{c_1} t) + A(1 - m(t))\cos(2\pi f_{c_2} t) \tag{7}$$

where $m(t)$ is a baseband signal that takes on values of either 0 or +1 during each bit interval. Note that in this form, FSK can be interpreted as the sum of two ASK signals.

Alternatively, FSK can be expressed as a frequency modulated signal of the form

$$s(t) = A\cos\left[2\pi f_c t + k_f \int_{\infty}^{t} m(\lambda)d\lambda\right] \tag{8}$$

where $m(t)$ is a baseband signal that takes on values ± 1 during each bit interval. The complex envelope is

$$g(t) = Ae^{j\theta(t)} \tag{9}$$

where

$$\theta(t) = k_f \int_{\infty}^{t} m(\lambda)d\lambda \tag{10}$$

4. Recall that for passband signal $s(t)$ having complex envelope $g(t)$,

$$s(t) = \Re\{g(t)e^{j2\pi f_c t}\} \tag{11}$$

where $g(t)$ is a baseband signal. If $g(t)$ has power spectral density $S_G(f)$ then $s(t)$ has power spectral density

$$S_S(f) = \frac{1}{4}\left[S_G(f - f_c) + S_G(-f - f_c)\right] \tag{12}$$

Therefore, the power spectral density of ASK, PSK, and FSK signals contain frequency shifted copies of the baseband signal power spectral density.